Although the Examiner has alleged difficulty with prior reasons for traversal already of record, the applicant believes those reasons to be sound and they are hereby incorporated by reference.

The Examiner is thanked for providing an extensive "response to arguments" section at pages 2-6 of the outstanding "final" Office Action. To expedite matters, this response and request for reconsideration is primarily directed towards further discussion of each of the Examiner's "response to arguments" allegations.

The "response" paragraph bridging pages 2-3 merely summarizes the Examiner's understanding of some features of applicant's earlier response -- which the Examiner finds not persuasive.

In the paragraph bridging pages 3 and 4, instead of referring to actual teachings of Wilson, the Examiner alleges (six times) that some feature is "obviously" found -- but without actual support from Wilson being identified.

The Examiner states that "applicant is reminded that the rejection only has to meet the claim language, not limits from the specification". In the light of this, the Examiner is reminded of some relevant words from claim 1 relating to the claimed encipher functional module:

> "forming means for receiving the signal to be enciphered and for outputting the signal as a sequence of data blocks, each having a first predetermined number of bits"

and

> "a plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means"

and

> "wherein each encipher functional module comprises
> a module input,
> a module output, and
> a respective data processing unit having a parallel input and a corresponding parallel output and being arranged to perform a respective reversible process upon a set of bits at its parallel input and to produce at its corresponding parallel output a corresponding enciphered set of bits,
> and is operable under the control of the configuring means to couple a respective predetermined set of the bits of a data block received at its module input to the parallel input of its data processing unit and to provide at its module output an enciphered data block in which said respective predetermined set of bits is replaced by the corresponding enciphered set of bits produced at the parallel output of its data processing unit."

The claimed invention therefore relates to a technique that is known in the art as "block enciphering". This means that an encipher module receives at its input an <u>n-bit</u> data block, enciphers it and outputs the enciphered data as an <u>n-bit</u> data block. This output data block is then applied to the input of another encipher module, and so on; the plurality of encipher modules being sequentially coupled.

The Examiner refers to Figure 1 of Wilson and states that he "believes that each of the boxes or group of boxes shown reads on functional modules" without ever identifying exactly what "box" or "boxes" are being referenced, the Examiner next alleges that each of the unidentified boxes is an encipher functional module. He then attempts to substantiate this belief by attempting to read applicant's definition of an encipher

functional module onto a part of Figure 1 in Wilson. The Examiner states his belief "that the shift register 18 along with the AND gates CA(sub(0)-CA(sub(63)) constitutes one functional module having parallel input and parallel output".

Actually, shift register 18 receives a serial data stream from buffer memory 22 under control of clock pulses from a clock pulse generator 24.

The Examiner also alleges that

> "the input into the shift register consist of the buffer memory data and a clock signal. There may also be an enable signal line not shown depending on the exact type of shift register used. These lines together reads [sic: read] on parallel inputs. The outputs from the AND gates reads on parallel outputs."
>
> The Examiner completes his analysis of this part of Wilson by alleging: "Obviously the data from buffer memory 18 (*sic*) is most likely serial in nature coming into the register 18. However, by using the shift register to shift the bits of data and the AND gates, the data is made parallel going into the next functional module. Obviously, the data going into each of the different input lines of module 10 are different. The AND operations done in the first functional unit (sic) are replacements of the input data bits going into the first functional unit (sic)".

Although the Examiner has not explicitly identified the input of the shift register as the module input, nor the AND gate outputs as the module output, it is assumed such was intended. The Examiner identifies them as a parallel input and a parallel output, but makes no reference to a data processing unit (as required in applicant's claim) which has such parallel input and parallel output. It would seem that the Examiner is assuming that the module input also constitutes the data processing unit input, and similarly for the

outputs, in which case the shift register and the AND gates must be assumed to constitute a data processing unit of that allegedly encipher functional module, and the "set of bits at its parallel input" must be loosely taken to be all the 64 bits clocked into the shift register (and forming a data block), and the "corresponding enciphered bits" must be the 64 AND gate outputs.

Assuming that applicant has thus correctly understood the Examiner's assumption and allegations there are fundamental problems with trying to read this onto the claimed data processing unit. For example, the claimed data processing unit is defined as performing a reversible process upon its input bits to produce a corresponding set of enciphered bits.

If one considers that an enciphered output is present at the AND gate outputs only when the encode command signal from encode command control 28 is at level "1", the encipher function would then be merely the unity function because the 64 AND gate outputs are identical to the data block contained in the shift register 18, and this would be so for every encode command signal, no matter what the bits of the input data block are. At all other times, when the encode command signal is at level "0", then the 64 AND gate outputs would all be at "0", and there would be no sensible meaning that could be given for an "encipher function."

The well known concept of reversibility in the field of encryption/enciphering is described in the specification, and includes the passage

> "Because the cipher units are reversible, the encipher apparatus and decipher apparatus can be constituted by a single apparatus. Thus, for duplex communication of an encrypted signal or for the storage of encrypted data for retrieval and decryption it is possible for the same cipher units to be used but in reverse order for deciphering."

This can also be seen in applicant's Figure 4 where the decipher apparatus 30 comprises the same set of cipher units 40a to 40d as in the encipher apparatus 10, but they are connected in reverse sequential order, i.e. from 40d to 40a. Thus, in order to decipher the enciphered signal, it is passed in sequence through the cipher units 40d to 40a, i.e. the signal to be deciphered is received at the input I of the cipher unit 40d, which provides an output signal to the input I of the cipher unit 40c, and so on. This must be equally true for the reversible operation of a single encipher functional module, i.e. if a data block (DB) is applied to the input of the module to produce at its output an enciphered data block (EDB), then applying the data block EDB to the input of the module will result in the output data block being identical to the original data block DB.

Thus, in respect of the first encipher functional module allegedly identified by the Examiner, it is submitted that the skilled person in the art would think that there is no merit in incorporating an encipher functional module whose encipher function is permanently merely the unity function, and indeed would not think of this part of Wilson

as anything other than its actual function of a serial to parallel converter with an admission control for gating the data block into the central memory 10.

Another important point which follows from the claimed feature of "reversible process", and taken together with the Examiner's contention of the central memory 10 constituting a second encipher functional module (and also that the third encipher functional module is constituted by the shift register 32 together with the XOR gates 30) is that it is inherent in claim 1 that the plurality of encipher functional modules sequentially coupled to operate sequentially on the sequence of data blocks from the forming means must have corresponding inputs and outputs so that a data block applied to the first module becomes a first-enciphered data block which is then applied to the second module to become a second-enciphered data block, and so on. If the number of bits in the output enciphered data block were not the same as for the input data block, it would not be practicable to perform the reversible deciphering.

In this regard, it must be noted that the alleged "second encipher functional module" 10 has a parallel input of 64 lines, but has a parallel output of 4096 lines. It is submitted that such a construction is untenable because an output data block of 4096 bits cannot be fed into the 64 input lines of central memory 10 to be deciphered.

The Examiner has not identified anything in Wilson's Figure 1 which might be the configuring means of claim 1, which determines, for an encipher functional module, the

respective predetermined set of the bits of a received data block which are coupled to its data processing unit. The Examiner has allegedly "identified" three encipher functional modules, so there needs to be, for each of these, a respective predetermined set of the bits of a received data block which are coupled to the respective data processing unit, and a configuring means which determines these predetermined sets of bits.

Presumably, although the Examiner is silent on this point, he might argue that the shift registers S together with their associated AND gates constitute the data processing unit of the alleged "encipher functional module" central memory 10, and that this unit has a 64 line parallel input connected to the input lines L, so that the "respective predetermined set of the bits of a received data block" for the central memory 10 is again all of the bits of the received data block. In which case, there is no proper construction and application of the words "respective predetermined set of the bits of a received data block" if for both the Examiner's first and second identified module "respective predetermined" is to be taken as all the received data block bits.

Another point on which the Examiner is silent is the identification in Wilson's Figure 1 of the above-mentioned "forming means" of claim 1. The applicant submits that the items 18, 22 to 24, 28 and AND gates CA(sub(0)-CA(sub(63)) in Figure 1 are a means for receiving the signal to be enciphered and for outputting the signal as a sequence of data blocks, each having a first predetermined number of bits. The encode command signals (every 64 clock pulses) produce at the AND gate outputs a sequence of

data blocks, each having 64 bits. The applicant submits that these items cannot at the same time constitute anything else in applicant's claim 1.

Turning now to Feistel and the Examiner's rebuttal comments, in the first paragraph on page 5 of the Office Action, the Examiner states that the applicant "does not believe Feistel discloses a random number generator of the claimed limitation", and "cannot identify where in the cited passages a pseudo-random number is encoded to provide respective descriptions of predetermined sets of bits of a data block as received at the respective module's input".

Again, because the Examiner has reminded applicant that the rejection only has to meet the claim language, it is best to have in mind the words from claim 71 relating to the claimed apparatus for generating a cipher design description:

> "a random or pseudo-random number generator; and encoding means arranged to receive a random or pseudo-random number generated by said generator and to encode that received number to provide at its output a cipher design description describing, for each of a plurality of cipher functional modules sequentially coupled to operate sequentially on a data block applied to the plurality of sequentially coupled cipher functional modules, a respective predetermined set of the bits of the data block as received at the respective module's input."

The Examiner identifies three passages from Feistel (column 3, lines 62-64; column 5, lines 6-20; and the last sentence of the abstract), and states "it is clear from the last sentence of the abstract that there is a pseudo-random number generator". He also

quotes from the last sentence of the abstract, and emboldens the words "**pseudo-random number generator**".

Column 3, lines 62-64 of Feistel actually reads:

> "The resultant cryptographic system is thus capable of providing either a very secure block cipher cryptogram or a much faster but somewhat less secure stream cipher cryptogram utilizing the system as a pseudo-random number generator."

Column 5, lines 6-20 of Feistel actually reads:

> "FIG. 1 is a broad functional block diagram of the present system and clearly shows the primary functional components thereof. The following description will relate generally to this figure. When operating in the block cipher mode, blocks of data are sequentially loaded into the main reconfiguration means which in the present embodiment is the Main Shift Register, the loading of said block is a function of one of said two user supplied keys (Key #2 in the FIG.). Subsequent to the loading operation, the contents of the Main Shift Register are reconfigured a predetermined number of times in combination with certain cryptographic transformation functions performed by the Transformation Element, shown in FIG. 1. Upon completion of said predetermined number of transformations the contents of the Main Shift Register (MSR) are gated directly out of the system as the enciphered or deciphered block of data.

The last sentence of the abstract of Feistel actually reads:

> "In the stream encipherment mode of operation the second key is entered in its entirety into the system where it is successively and continuously transformed as a function of said first key whereby the function of said system becomes a pseudo-random number generator whose output is serially combined with the raw data to form the stream enciphered cryptogram."

The only other passages relating to "random" found in Feistel are:

Column 2, lines 32-35 which reads " . . . U.S. Pat. No. 3,506,783 filed June 12, 1967 which discloses a means for generating the key-material which gives a very long pseudo-random sequence."

Column 3, lines 2-18 which reads "Conversely, as stated previously with the stream-cipher systems utilizing some sort of a stream-generator, either the complete random number stream or key must be known at both the sending and receiving ends or alternatively some form of known psuedo-random (*sic*) number generator must be used. It is generally considered impractical to have a complete secret random number key. Accordingly, when stream encipherment is to be accomplished the prior art normally utilizes some sort of pseudo-random number generator. The primary advantage of stream encipherment is its speed, i.e., the message is flowed serially through the system and the data stream combined in a known transformation with the random number generator as by a modulo-2 addition which may be repeated at the other end for decryption with maximum speed."

Column 4, lines 12-15 which reads "in one case [*i.e. block mode*], the output of said system is a block cipher cryptogram and in the stream mode the output comprises a sequential psuedo-random (*sic*) number stream."

Column 4, line 66 to column 5, line 2 which reads "When operating in the stream mode, the hardware functions primarily as psuedo-random (*sic*) key generator, which is combined serially with the input data stream in a modulo-2 adder."

Column 5, line 67 to column 6, line 10 which reads ". . . the overall system operates essentially as a psuedo-random number generator, wherein the contents of the Main Shift Register are continuously altered by the Transformation Element utilizing the operations of substitution, permutation or transposition as well as modulo-2 addition to produce the aforesaid psuedo-random number stream. Each time the Main Shift Register is shifted one bit position, as will be apparent from the following description, there is an output bit which is modulo-2 added to a bit of the data stream regardless of of (*sic*) whether said data stream is being enciphered or deciphered."

Column 8, lines 20-26 which reads "AND-gate 7 is enabled in the stream mode, via a control signal from switch SW1(5), and thereby gates the input message stream to adder 18 where said message stream is summed, modulo-2, with the pseudo-random binary bit-stream fed to Adder 18 from

the Main Shift Register 14. The system output, i.e., the "Processed Output", is taken from the output of Adder 18."

Column 12, lines 8-32 which reads "This completes the description of the stream encipherment mode. It will be noted that the cipher/decipher switch SW-2 need never be interrogated in this mode. This is because regardless of encipherment or decipherment the same pseudo-random number stream will be produced by the present system and will always provide the identical input to the modulo-2 adder 18. It will readily be appreciated that after a first modulo-2 addition with a known binary stream, with a second known bianary (*sic*) stream, that a resultant third binary stream will be produced by such an adder. In order to then produce one of the original binary streams it is only necessary to have the combined output and one of the originals. In the present instance the pseudo-random number output stream from the MSR is duplicated at the receiving end and by producing this pseudo-random number stream and mixing or decoding in the modulo-2 adder 18 with the encoded data stream, the original clear-text binary stream is produced. Thus, in the stream mode of operation, encipherment and decipherment operations are essentially identical. The only difference is that in the encipherment case the received message is clear and in the decipherment case the received message would have been previously enciphered or be 'ciphertext'."

Column 19, lines 31-35 which reads "However, the underlying concept of utilizing essentially the same hardware to generate a pseudo-random number stream for use as a stream cryptographic system or as a rather complex block cryptographic system is considered to be basically unique."

Claim 1 includes "means for selectively utilizing the complete contents of said main reconfiguration means as a full cryptographically transformed block of data in block mode operation, or a pseudo-random number stream which is combined serially in an inverse mathematical function with said incoming data set in serial mode operation."

Claim 7 includes "means operable during stream encipherment mode for causing said shift register to be sequentially shifted in a circular mode and

> wherein a selected bit is extracted on each shift operation to form a series of output bits which is used as a pseudo-random number stream."
>
> Claim 8 includes "including a modulo-2 adder for combining said pseudo-random number stream with said incoming data set to be cryptographically transformed and wherein the output of said modulo-2 adder comprises said cryptographically transformed data stream."
>
> Claim 11 includes "means for selectively utilizing the output of said main reconfiguration means as a full cryptographically transformed block of data in the block mode of operation or a pseudo-random number stream which is to be cryptographically combined with said incoming data set when in the stream mode of operation."
>
> Claim 13 includes "means operable during stream mode operation whereby the output of a predetermined bit position of said main reconfiguration means is utilized as the output random number bit stream to be combined with said incoming data set and further including modulo-2 adder means for combining said pseudo-random number bit stream with said incoming data set to produce a cryptographically transformed data stream output."

Thus, it will be understood that each and every reference in Feistel to a pseudo-random number generator is in connection with stream mode encipherment and decipherment, i.e. modulo-2 addition of the pseudo-random number with the original clear-text binary stream to produce an encoded data stream, and modulo-2 addition with the encoded data stream to produce the original clear-text binary stream.

It should now be understood that Feistel's pseudo-random number is the sequence of output pulses that is produced from the MSR as the process cycles around blocks 4, 5, 6 and 7 of Figure 3 until block 7 determines an end of message condition.

The Feistel sequence of output pulses (pseudo-random number) is produced by initially clocking the 64 bit key 2 into the MSR (block 3); and then repetitively loading

the MSR destination fields under the control of the output bit of the KSR 1; shifting the MSR one position to the right; and then shifting the KSR 1 (which is end-around connected) one position to the right to obtain the next bit for controlling the "load MSR DF" operation. Feistel does not suggest or require that the 64 bit key 2 (or the 64 bit key 1) is a pseudo-random number (PRN). The only use of the PRN is in the Feistel stream mode as an input to the modulo-2 adder 18 for adding bit-by-bit with the input data stream in known fashion. The PRN is not used in Feistel's block mode.

Feistel discloses that his two encipher modes are not just two different ways of producing the same enciphered version of input data. On the contrary, he states at column 3, lines 20-34 "However, in many communication and/or closely related computer systems where differing security levels exist, it would be a great advantage to be able to utilize full block ciphering techniques for highly secure data transmissions and stream enciphering techniques for data transmissions having a requirement of lower security.

An example of such a system might be in a cash issuing or banking terminal wherein the personal identification of the person seeking to obtain money or credit must be of the highest security to insure proper identification while the actual data message transmission could be at a lower level of security, but wherein some security or secrecy might be desired to maintain the integrity of the data being transmitted.

In other words, the two encipher processes are different. The PRN in the stream mode does not appear in the block mode.

In the Feistel block mode, he does not have a physical sequence of encipher modules, each performing a respective enciphering of a data block received at its input and passing on its enciphered data block to the input of the following encipher module. Feistel accepts a 64 bit data block (in serial form) enciphers it and outputs the enciphered data block (also in serial form).

If the "cycling and transformation" of the MSR under the control of the 64 bit key 1 were to be considered equivalent to the sequential processing of a data block by 64 sequentially coupled encipher functional modules, then the operation of such a module would be input bit 0 transferred (by the shift MSR operation) without transformation as output bit 1; input bit 1 similarly transferred without transformation as output bit 2, etc down to input bit 55 transferred without transformation as output bit 56; then input bit 56 transformed/enciphered by a "load MSR DF" operation and transferred to be output bit 56, etc down to input bit 63 transformed/enciphered by a "load MSR DF" operation and transferred via the MSR Multiplexer 10 to be shifted into the MSR as a new input bit 0.

In this case, the "respective set of bits of the received data block" for each of these "sequential" modules would always be bits 56 to 63, i.e. the four-bit destination fields C and D. In other words, these destination fields C and D are not selected in accordance

with a cipher design description derived by encoding a PRN, but chosen by the designer of the transformation process. It should now be understood that the action of shifting the MSR to the right and producing a PRN stream at its output is not the same or even equivalent to the recited requirement of applicant's claim 71 of "a cipher design description describing, for each of a plurality of cipher functional modules sequentially coupled to operate sequentially on a data block applied to the plurality of sequentially coupled cipher functional modules, a respective predetermined set of the bits of the data block as received at the respective module's input".

The Feistel MSR stream mode output is not related in any way to block mode enciphering.

As mentioned above, the only references in Feistel to "pseudo-random number" are in respect of the function of the system in stream mode to be a pseudo-random number generator, i.e. that the output of the MSR is a pseudo-random number.

In applicant's previous response, applicant did <u>not</u> take the position that Feistel did not disclose a pseudo-random number generator *per se* (Feistel's stream mode functions as a pseudo-random number generator), but rather that Feistel could not read on claim 71 because the output of the pseudo-random number generator was not coupled to an encoding means which encoded the PRN from the MRS to provide "at the output of the encoding means a cipher design description describing, for each of a plurality of cipher

functional modules sequentially coupled to operate sequentially on a data block applied to the plurality of sequentially coupled cipher functional modules, a respective predetermined set of the bits of the data block as received at the respective module's input".

In the same paragraph on page 5 of the Office Action, the Examiner asserts that the second key (key 2) is "obviously" a PRN generated by a generator (but this is not shown or mentioned in Feistel), and that transformation of this second key reads on the pseudo-random number being encoded. In other words that means operating in accordance with blocks 4, 5, 6 and 7 encode the key 2 to produce at the MSR output a cipher design description.

The Examiner goes on to assert "As this number is used for encryption of data, it must provide respective descriptions of predetermined sets of bits of data block as received at the respective module's input".

Taking the Examiner's reference to "this number" as meaning the second key, the successive and continuous transformation of the second key will produce Feistel's pseudo-random number output from the MSR. The Examiner now asserts that the "encoded" second key (*i.e. the output from the MSR*) "must provide respective descriptions of predetermined sets of bits of a data block as received at the respective module's input".

However, in fact, the output from the MSR is merely module-2 added via adder 18 with the input message stream. To make a connection between the output of the MSR and a cipher design description describing the "respective predetermined sets of received data blocks", the Examiner has attempted to identify in Feistel "a plurality of cipher functional modules sequentially coupled to operate sequentially on a data block applied to the plurality of sequentially coupled cipher functional modules". He asserts that Figure 1 shows such a plurality of sequentially coupled cipher functional modules. He states that "each of the boxes [*in Figure 1*] represents modules. Obviously the output from one module is connected to the input of another module, so there is cascading of modules.".

However, Figure 1 shows five boxes. Two of the boxes signify the two user supplied keys (KEY#1 and KEY#2). In practice, the keys will be stored in memory, so that it might be considered that the memory has an input for receiving the key, and an output for reading out the key. However, since the output is identical to the input, it is submitted that neither of these boxes would be considered by a skilled person in the art as a cipher functional module for operating on a data block received at the module input.

Another of the boxes is the STREAM/BLOCK LOGIC box which functions to change the operation of the system between stream mode and block mode. Further description of the two modes need not be included here, but it should now be clear that the input to this box is the input message stream, and the output of this box (shown in Figure 1 as the input to the MSR) is either the input message stream modulo-2 added with

key 2 (in block mode) or key 2 itself (in stream mode). It will be noted that Figure 1 does not accurately represent the stream mode connections as shown in Figure 2. Figure 1 shows the output of this box being fed as an input to the MSR, but in stream mode the input message stream is not applied as an input to the MSR, but is connected as one input of the modulo-2 adder 18 which receives the output of the MSR at its other input.

As just stated, the input to this box is the input message stream, and not a data block. The creation of a data block (i.e. a 64 bit block of data) is performed within the box by loading the value 63 from the ROM 22 into the operation counter 25, and controlling the modulo-2 addition of bits of the input message stream with the output of KSR2 by clock pulses until the operation counter 25 counts down 64 clock pulses and eventually applies zero to input 3 of the input multiplexer 20. If it considered that the output data block of this box (via input 1 of the multiplexer 10) is an input data block enciphered by the modulo-2 addition with key 2, this leaves the question as to what is the design description relating to this module and what is the respective predetermined set of the bits of the input data block.

The Examiner alleges that the encoding means of claim 71 is the MSR - - the second key is entered in its entirety into the system where it is successively and continuously transformed as a function of said first key whereby the function of said system becomes a pseudo-random number generator - - The transformation of the second

key, which is obviously a pseudo-random number as most keys are, reads on the pseudo-random number being encoded.

However, the "successive and continuous transformation of the second key" being referred to by the Examiner, occurs only in the stream mode. The Examiner thus erroneously alleges that the output of the MSR in stream mode is the encoded PRN of applicant's claim 71. Feistel uses this output for one reason only: for modulo-2 addition with the input data stream in stream mode. The second key will not be encoded by the MSR in block mode, i.e. in block mode encipherment the second key is NOT entered in its entirety into the system to be successively and continuously transformed.

The second key is thus being alleged by the Examiner to be (in block mode) the encipher function of a module (the STREAM/BLOCK LOGIC box) and, at the same time, a PRN which (in stream mode) is encoded (by the MSR) to provide at the MSR output, a cipher design description describing, inter alia, a respective predetermined set of the bits of the input data block. The "successive and continuous transformation" of key 2 cannot be given a meaning for block mode.

A fourth of the boxes is the TRANSFORMATION ELEMENT box which is essentially items 12, 13, 15, 16, 17 and 19 as shown in Figure 2. This does not receive a data block at its input and produce at its output an enciphered version of a received data block. The input from key 1 is applied one bit at a time to the Permutation Control Line

for applying the permutation control variable, P, to the multiplexers 15 and 19, but this box (contrary to what is shown in Figure 1) also receives sixteen inputs in parallel from the source and destination fields A, B, C and D of the MSR. The output from this box is actually the eight bits output in parallel from the adders 16 and 17. Thus there is no concept here of block enciphering, i.e. there is no concept here of an n-bit input data block and an n-bit output data block.

The last of the boxes is the MAIN SHIFT REGISTER which is item 14 of Figure 2. This alone of the boxes can be said to be an encipher module that receives a 64-bit data block (albeit not purely of the input message stream but 64 bits of message data that have been modulo-2 added with key 2) and will provide at its output an enciphered 64-bit data block.

Therefore, at best, Figure 1 shows cascaded "modules" STREAM/BLOCK LOGIC and MAIN SHIFT REGISTER, but it is not possible to identify for each of these modules where "the transformation of the second key" becomes a cipher design description, nor where for each of these modules there is a description of a respective predetermined set of the bits of its received data block.

Considering now, the Examiner's statement that "The transformation of the data are functions of the keys, therefore the <u>keys</u> provide a description of respective predetermined set of the bits as received at a module input", and the passage at column 5,

lines 6-20. The Examiner seems to allege by implication that key 1 is also a PRN, but has not stated this explicitly.

This passage is exclusively related to the "block cipher mode", which as mentioned above is a distinctly different and alternative operation of the system of Figure 1. In block mode, a 64-bit block of input data is modulo-2 added with the 64-bit key 2 and loaded into the MSR. The content of the MSR is now reconfigured (64 times) and transformed in a particular manner using the 64-bit key 1.

It would seem that the Examiner is taking the position that key 2 provides a description of a set of the bits of the 64-bit block of input data received at the input of the STREAM/BLOCK LOGIC module, and that key 1 provides a description of a set of the bits of the 64-bit block of input data received at the input of the MSR module. What would be the nature of this description? Perhaps the set of bits would be those in the same bit positions as the ones (or zeros) in key 2.

At the stage of producing the modulo-2 added data block, key 2 is used in its original state. It has not undergone any encoding. As the Examiner has already identified key 2 as the PRN of our claim 71, and the <u>serial mode</u> transformation as the encoding step of our claim 71, there is not present an encoded form of key 2 which describes a set of bits of an input data block received by the STREAM/BLOCK LOGIC module.

Furthermore, there is no successive and continuous transformation of key 1 corresponding to the "encoding" of key 2, so there is no encoding of key 1. Still further, our claim 71 requires that the a (single) cipher design description describing all the respective sets of bits for the plurality of modules results from the encoding of a (single) PRN, not that a first PRN describes a set of bits for a first module and a second PRN describes a set of bits for a second module.

In truth, although Feistel has described an encipherment apparatus which can be switched between block and stream modes, he does not disclose the applicant's claimed invention (e.g., see any independent claim herein), and the Examiner is, with the benefit of hindsight, identifying features of the two modes and attempting to combine them with statements that do not agree precisely with the wording of the pending claims.

Accordingly, this entire application is now believed to be in allowable condition and a formal Notice to that effect is respectfully solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: Larry S. Nixon
Larry S. Nixon
Reg. No. 25,640

LSN:vc
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100